CLAIMS:

1. A low weight encoding circuit comprising:

a current balance tester arranged to test whether a predetermined number of data bits is current balanced;

a current balance encoder and decode bit generator arranged to encode data bits and generate encoded data and corresponding decode bits if said predetermined number of data bits is not current balanced; and

a latch arranged to latch either the data bits, via an I/O bus, if said predetermined number of data bits is current balanced or the encoded data and corresponding decode bits, via the I/O bus, if said predetermined number of data bits is not current balanced.

2. The low weight encoding circuit as claimed in claim 1, wherein said predetermined number of data bits corresponds to 4 bits or a data nibble from the I/O bus supporting at least 32-bit or 64-bit data transactions.

3. The low weight encoding circuit as claimed in claim 2, wherein said current balance tester receives the data nibble at one bus transition, determines if the data nibble is already current balanced (i.e., two bits high and two bits low), and if the data nibble is already current balanced, allows the data nibble to latch onto the I/O bus at the next bus transition.

4. The low weight encoding circuit as claimed in claim 2, wherein said current balance encoder and decode bit generator encodes the data nibble to ensure that the data nibble is current balanced prior to being driven onto the I/O bus and generates therefrom encoded data and corresponding decode bits, and allows the encoded data and corresponding decode bits to latch onto the I/O bus at the next bus transition.

5. The low weight encoding circuit as claimed in claim 2, wherein said current balance tester is implemented using combinations of Boolean logics including, but not limited to, XOR and XNOR logic gates arranged to test the data nibble for balanced current drawn using A_0, A_1, A_2 and A_3 as parameters representing individual bits in the data nibble of four (4) data bits and the following equations: (1) $F_1 = \overline{A_0 \oplus A_1}$, (2) $F_2 = \overline{A_2 \oplus A_3}$, (3) $F_3 = \overline{F_1 \oplus F_2}$, (4) $F_4 = A_0 \oplus A_1 \oplus A_2 \oplus A_3$, and (5) $F_3 \bullet F_4 = 1$, and if $F_3 \bullet F_4 = 1$, then the data nibble is current balanced.

6. The low weight encoding circuit as claimed in claim 2, wherein said current balance tester is implemented using a look-up TABLE containing logic functions F_1, F_2, F_3 and F_4 pre-assigned to different combinations of the input data nibble to test whether the data nibble is current balanced.

7. The low weight encoding circuit as claimed in claim 5, wherein said current balance encoder and decode bit generator is implemented using combinations of Boolean logics including, but not limited to, XOR and XNOR logic gates, initial bit patterns A_0, A_1, A_2 and A_3 as parameters representing individual bits in the data nibble of four (4) data bits and the following equations: (1) $\overline{A_0 \oplus A_1 \oplus A_2} \bullet \overline{F_3} = 1$, (2) $\overline{F_3} \bullet F_4 = 1$, and (3) $F_4 = 1$.

8. The low weight encoding circuit as claimed in claim 7, wherein said current balance encoder and decode bit generator is configured to determine if $\overline{A_0 \oplus A_1 \oplus A_2} \bullet \overline{F_3} = 1$, and if $\overline{A_0 \oplus A_1 \oplus A_2} \bullet \overline{F_3} = 1$, invert bit A_0 and bit shift the data nibble to the left, carry bit A_0 into the A_3 position, and generate encoded data and corresponding decode bits.

9. The low weight encoding circuit as claimed in claim 8, wherein said current balance encoder and decode bit generator is configured to determine if $\overline{F_3} \bullet F_4 = 1$, and if $\overline{F_3} \bullet F_4 = 1$, invert bit A_3 and bit shift the nibble to the right, carry bit A_3 into the A_0 position, and

generate encoded data and corresponding decode bits.

10. The low weight encoding circuit as claimed in claim 9, wherein said current balance encoder and decode bit generator is configured to determine if $F_4 = 1$ and if $F_4 = 1$, invert bits A_1 and A_3, and generate encoded data and corresponding decode bits.

11. The low weight encoding circuit as claimed in claim 2, wherein said current balance encoder and decode bit generator is implemented using a look-up TABLE containing encoded nibbles are pre-assigned to different combinations of input data nibbles A_0, A_1, A_2 and A_3 that need to be encoded.

12. An electronic system, comprising:

a power supply;

an active circuit including a plurality of I/O cells which serve as switching transistors; and

a power delivery system arranged to deliver a transient current from the power supply to the I/O cells of the active circuit, via an I/O bus, said power delivery system comprising a low weight encoding circuit arranged to encode data bits sent out on the I/O bus such that minimal current drawn is achieved to minimize signal and timing distortions.

13. The electronic system as claimed in claim 12, wherein said low weight encoding

circuit comprises:

a current balance tester arranged to test whether a predetermined number of data bits is current balanced;

a current balance encoder and decode bit generator arranged to encode data bits and generate encoded data and corresponding decode bits if said predetermined number of data bits is not current balanced; and

a latch arranged to latch either the data bits, via the I/O bus, if said predetermined number of data bits is current balanced or the encoded data and corresponding decode bits, via the I/O bus, if said predetermined number of data bits is not current balanced.

14. The electronic system as claimed in claim 13, wherein said predetermined number of data bits corresponds to 4 bits or a data nibble from the I/O bus supporting at least 32-bit or 64-bit data transactions.

15. The electronic system as claimed in claim 14, wherein said current balance tester is implemented using combinations of Boolean logics including, but not limited to, XOR and XNOR logic gates arranged to test the data nibble for balanced current drawn using A_0, A_1, A_2 and A_3 as parameters representing individual bits in the data nibble of four (4) data bits and the following equations: (1) $F_1 = \overline{A_0 \oplus A_1}$, (2) $F_2 = \overline{A_2 \oplus A_3}$, (3) $F_3 = \overline{F_1 \oplus F_2}$, (4)

$F_4 = A_0 \oplus A_1 \oplus A_2 \oplus A_3$, and (5) $F_3 \bullet F_4 = 1$, and if $F_3 \bullet F_4 = 1$, then the data nibble is current balanced.

16. The electronic system as claimed in claim 14, wherein said current balance encoder and decode bit generator is implemented using combinations of Boolean logics including, but not limited to, XOR and XNOR logic gates, initial bit patterns A_0, A_1, A_2 and A_3 as parameters representing individual bits in the data nibble of four (4) data bits and the following equations: (1) $\overline{A_0 \oplus A_1 \oplus A_2} \bullet \overline{F_3} = 1$, (2) $\overline{F_3} \bullet F_4 = 1$, and (3) $F_4 = 1$.

17. The electronic system as claimed in claim 16, wherein said current balance encoder and decode bit generator is configured to:

invert, if $\overline{A_0 \oplus A_1 \oplus A_2} \bullet \overline{F_3} = 1$, bit A_0 and bit shift the data nibble to the left, carry bit A_0 into the A_3 position, and generate encoded data and corresponding decode bits;

invert, if $\overline{F_3} \bullet F_4 = 1$, bit A_3 and bit shift the nibble to the right, carry bit A_3 into the A_0 position, and generate encoded data and corresponding decode bits; and

invert, if $F_4 = 1$, bits A_1 and A_3, and generate encoded data and corresponding decode

bits.

18. The electronic system as claimed in claim 16, wherein said current balance encoder and decode bit generator is implemented using a look-up TABLE containing encoded nibbles are pre-assigned to different combinations of input data nibbles A_0, A_1, A_2 and A_3 that need to be encoded.

19. A method of encoding data in a power delivery system, comprising:

determining whether a predetermined number of data bits is current balanced;

if said predetermined number of data bits is current balanced, driving said predetermined number of data bits onto an I/O bus; and

if said predetermined number of data bits is not current balanced, encoding said predetermined number of data bits to ensure that said predetermined number of data bits is current balanced prior to being driven onto the I/O bus along with corresponding decode bits as to minimize signal and timing distortions.

20. The method as claimed in claim 19, wherein said predetermined number of data bits corresponds to 4 bits or a data nibble from the I/O bus supporting at least 32-bit or 64-bit data transactions.